Database Design for a Book Recommendation Engine

Thejas Raju Ravi Govind Raju (18204398)
thejasraju96@gmail.com

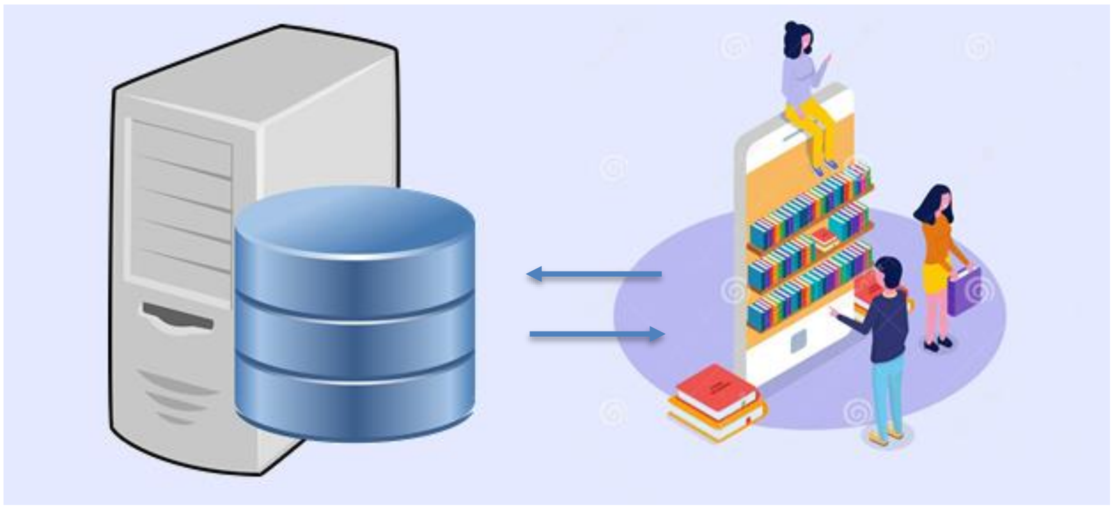


Table of Contents

List of Figures

1. Introduction

The project is intended to design a database system using the MySQL framework which caters to support the working of a bookseller's catalogue and recommendation system. The database forms the server side of the client/server architecture and the book recommendation application makes up the client side. The application communicates with the server to insert, modify or retrieve data which is present in the database through queries and procedures. The communication between the client and server takes place through a series of transactions.

The data being used to provide recommendation consists of tables which contain data ranging from the inventory of books available to keeping track of customer information and the orders placed by them.

The books inventory table is divided into a subsequent group of tables which are consistent with the normalization laws. The tables: 'authors', 'themes', and 'qualities' consist of a column containing ISBNs which helps in identification of the book and a rank column to organize books based on their corresponding author-rank or theme-rank or quality-rank. Another table containing the book titles and their corresponding ISBNs is used to store the name of the books. The titles table consist of 588 entries, one for each book present in the inventory. However, the authors, themes, and qualities consist of 604, 2699, and 3656 records each. This can be attributed to the fact that a book may have more than one authors and the book can feature multiple themes and qualities.

The 'customers' table on the other hand consists of columns to store the first and last names of the customer. In addition to these columns, a separate column customerID is used to assign a unique integer to each column which helps in easy identification and keeping track of the customer across various other tables. The number of records in the customers table depends on the number of recorded users of the client-application. This can be very huge as the volume of traffic across an online book-sellers website can be high.

Other information related to the customer such as their age, gender, address, country of residence, and profession are stored in individual tables which are identified using the primary key-customerID.

To obtain information about a particular customer, the customerID can be matched in each table containing customer details and retrieve data appropriately.

The orders placed by each customer is recorded in an 'orders' table which stores the customerID and assigns a unique orderID to each order that is placed. The details of the books that are ordered is stored in a different table called 'ordersISBN' which keeps track of the books order using their ISBNs. The ISBN of the book ordered and the order which they belong to are stored respectively in the ISBN and OID columns of the ordersISBN table.

To obtain details about a book that is ordered, the ISBN which corresponds to the specific orderID can be used to perform a look-up to the titles, authors, themes, and qualities tables.

In order to draw more information regarding the customer who has placed an order of a specific book, the orderID which corresponds to the ISBN can be used to perform a look-up to the orders table which provides the customerID recorded to the corresponding orderID. This customerID can be used to retrieve further details such as name, address, age, and profession from the customers table.

The database design plays a major role in supporting the application as it acts as a backbone to the recommending engine. It caters to the application by providing storage and fast access to records. The database is not in itself a book-recommending engine but rather supports the recommending engine to function seamlessly. The process of making recommendations can be broadly categorized into two: item-based filtering and user-based collaborative filtering. (Finnerty, 2017)

In the item-based filtering method, recommendations are made in accordance to the previous purchases of the particular user. Customer preferences such as favored authors, book-themes, and book-qualities can be understood by examining previous purchases which are recorded in the database. For example, if a customer previously purchased a book authored by Agatha Christie, this information is taken into account by looking up the orders and authors table after which other books by Agatha Christie can be recommended to the customer.

In the user-based collaborative filtering method, recommendations are made after examining other users with similar characteristics. Here, a set of recommendations are made to the particular customer based on the preference of customers with a similar profile. For example, a customer can be receive recommendations based on his profession. If the customer is an actor, they receive recommendations after examining the purchases of other customers who are actors by profession. This can be facilitated in the database by recording order details of all customers and providing insights to their personal details which can help the recommending engine in examining the similar features (if any) that the customers possess.

The presence of individual tables containing data related to the themes and qualities of books help in recommendation to a new customer who can be asked to choose from a list of themes and qualities in the client application and process it to provide an appropriate set of books to choose from.

2. Database Plan: A Schematic View

The database consists of multiple tables which are in relation to each other and a few other individual tables which when combined together cater to the needs of a book recommendation engine.

The customers table forms an integral part of the database as it maintains the data related to customers who use the recommendation engine. It consists of fields firstName and lastName. Each record in this table is identified using a unique customerID (CID). The CID column is of type integer which increments automatically for each record inserted and is identified as the primary key of the customers table as it contains unique values for each record. The fields firstName and lastName are declared as type varchar as they hold textual data.

The orders table consists of the columns orderID (OID) and customerID (CID). Both fields are defined as type integer where the order value increases automatically for each record inserted and is defined as the primary key of the table. The CID column derives its values from the customers table and is defined as a foreign key. This is done so to ensure that orders can be traced back to valid users whose credentials are available in the database. In this table, each order placed is given a unique orderID and the CID of the customer who placed the order is stored in the corresponding field.

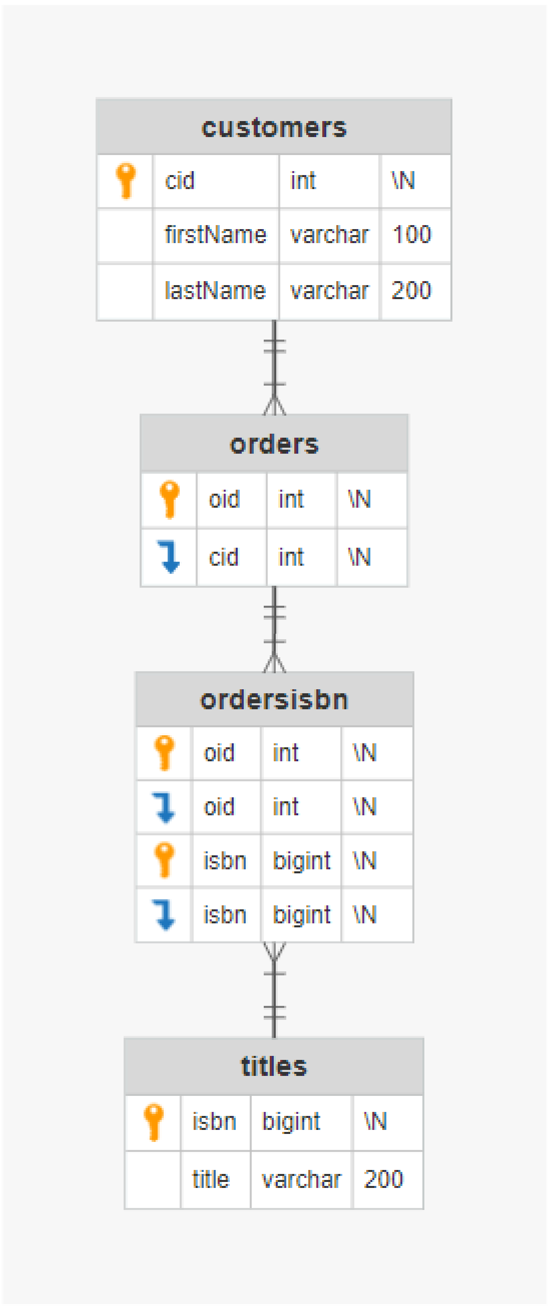


Figure 1: Entity-Relationship Diagram (a)

The relationship between the orders table and the customers table is such that there can be one or more orders placed by the same customer with each order being recorded independently. If a customer is yet to place an order, there is no record present in the orders table and hence the one or more relation exists. However, each order placed by a customer is recorded independently i.e. each individual order can be associated with only one customer and hence one and only one type of relation exists between customers and orders table.

The ordersISBN table consists of fields orderID (OID) and ISBN. The OID column is declared as of type integer whereas the ISBN column is defined as type bigint. In this table, both the fields make up a combined primary key. This is adopted as a single order can consist of multiple books and hence the same OID can be used against more than one ISBN values. The OID column refers to the orders table to derive its values to ensure that only a valid order is recorded. The ISBN column refers to the column present in the titles table to ensure that a valid book is being ordered.

The orders table and the ordersISBN table share a one and only one type relation where there exists only one unique OID for each order placed by a customer. For example, a customer placing an order is assigned a unique orderID and this orderID changes with every order placed even if it is by the same customer. However, the ordersISBN and orders table have one or many type relation. For example, a customer may order multiple books within the same order. Here, a single OID has more than one corresponding ISBN values justifying the one or more relationship.

The titles table consists of fields ISBN and the book-title where ISBN is used to uniquely identify a book. The ISBN field is of type bigint and the title field is of type varchar as it holds textual data.

The titles table and the ordersISBN table are connected mainly to authenticate that a valid book that exists in a database is being ordered. The ordersISBN table is related to the authors table such that there can be more than one orders which contain the same ISBN. For example, more than one customers can order the same book. However, the authors table defines that each order can be related to only one ISBN value that is present in the ordersISBN table. The database is designed such that each customer orders a single copy of the book which justifies this relationship.

The genderID table holds two columns gID and gender which specify the genderID to be assigned and the gender to which it is assigned respectively. The gender male is assigned value of '1' and female is identified by the value '2'. This is done to keep in line with normalization laws that govern the database.

The genders table holds two columns customerID and gender. In this table, the gender is stored as a numeric value whose key can be found in the genderID table. The customerID can be used to uniquely identify each record within the genders table and the gender column (of the genders table) refers to the genderID column (of the genderID) to derive its appropriate value.

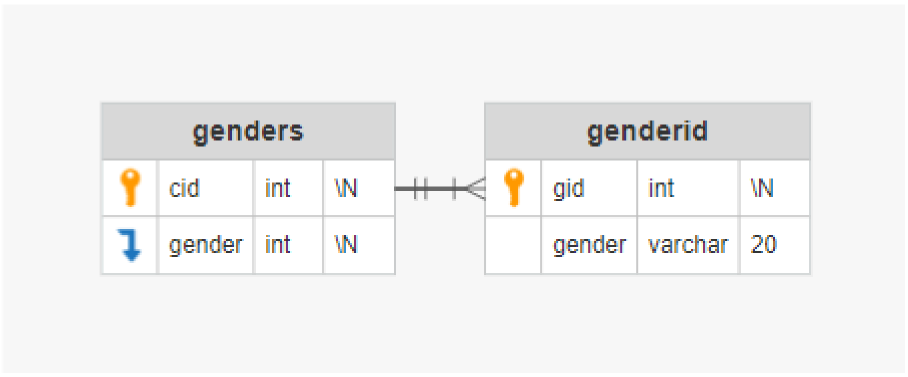
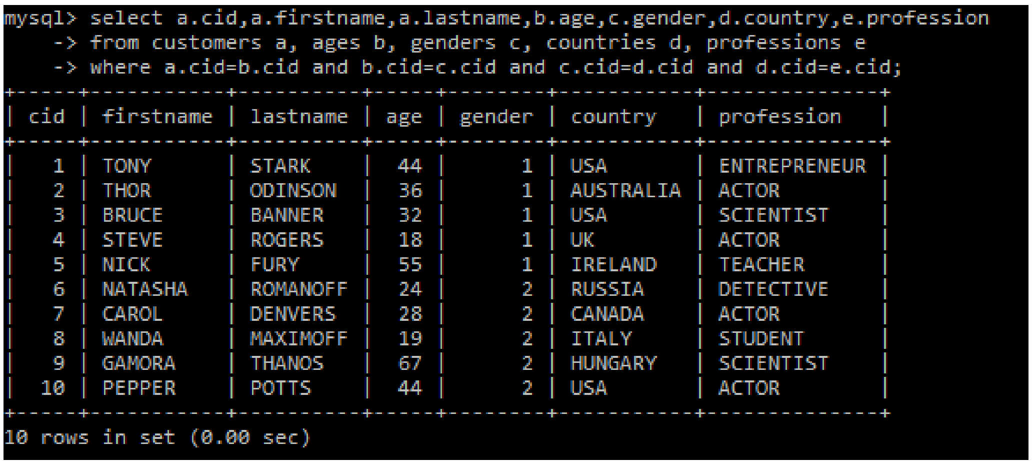


Figure 2: Entity-Relationship Diagram (b)

The relation between the genders table and genderID table is such that each customer can belong to only one gender type whereas there can be more than one customers who belong to the same gender. For example, there can be two customers who are males and each of them are identified using their respective customerID. But a customer can be either male or female and not both.

3. Database Structure: A Normalized View

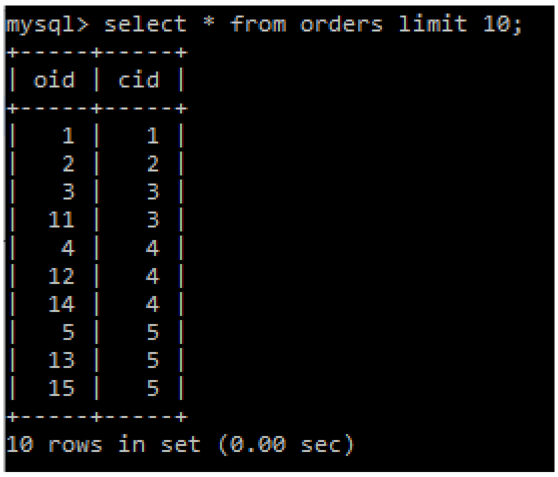
The customers table is an integral part of the database as it generates the customerID to be used across other tables which hold data relevant a customer. For example, the customerID from the customers table is used to associate orders placed by a particular customer and other information such as age, gender, address, and profession.



```
mysql> select a.cid,a.firstname,a.lastname,b.age,c.gender,d.country,e.profession
    -> from customers a, ages b, genders c, countries d, professions e
    -> where a.cid=b.cid and b.cid=c.cid and c.cid=d.cid and d.cid=e.cid;
+-----+-----------+----------+-----+--------+-----------+--------------+
| cid | firstname | lastname | age | gender | country   | profession   |
+-----+-----------+----------+-----+--------+-----------+--------------+
|   1 | TONY      | STARK    |  44 |      1 | USA       | ENTREPRENEUR |
|   2 | THOR      | ODINSON  |  36 |      1 | AUSTRALIA | ACTOR        |
|   3 | BRUCE     | BANNER   |  32 |      1 | USA       | SCIENTIST    |
|   4 | STEVE     | ROGERS   |  18 |      1 | UK        | ACTOR        |
|   5 | NICK      | FURY     |  55 |      1 | IRELAND   | TEACHER      |
|   6 | NATASHA   | ROMANOFF |  24 |      2 | RUSSIA    | DETECTIVE    |
|   7 | CAROL     | DENVERS  |  28 |      2 | CANADA    | ACTOR        |
|   8 | WANDA     | MAXIMOFF |  19 |      2 | ITALY     | STUDENT      |
|   9 | GAMORA    | THANOS   |  67 |      2 | HUNGARY   | SCIENTIST    |
|  10 | PEPPER    | POTTS    |  44 |      2 | USA       | ACTOR        |
+-----+-----------+----------+-----+--------+-----------+--------------+
10 rows in set (0.00 sec)
```

Figure 3: Customer Information obtained by matching CID

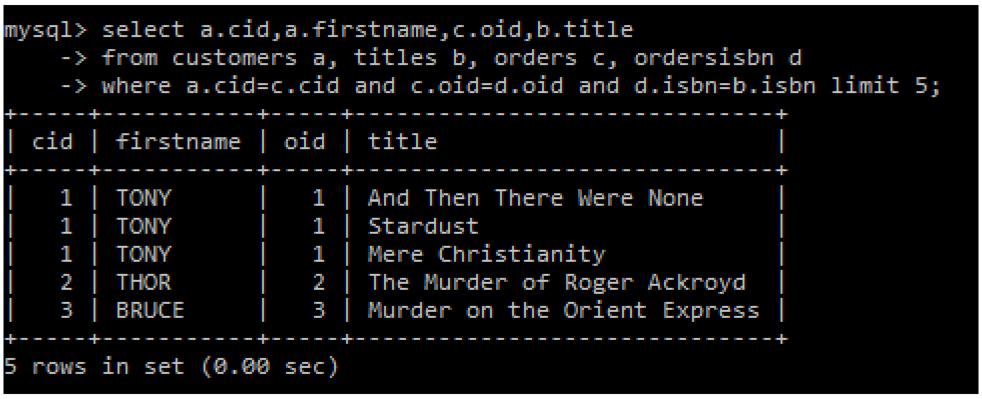
The orders table is essential as it is used to define an orderID to identify each order. The orderID is recorded against the customerID of the customer who has placed the order. This orderID is used across the database to obtain information about books which make up the order. This table can be used to trace the information of the customer who has placed the order using the customerID and the details of the order that has been placed using the orderID.



```
mysql> select * from orders limit 10;
+-----+-----+
| oid | cid |
+-----+-----+
|   1 |   1 |
|   2 |   2 |
|   3 |   3 |
|  11 |   3 |
|   4 |   4 |
|  12 |   4 |
|  14 |   4 |
|   5 |   5 |
|  13 |   5 |
|  15 |   5 |
+-----+-----+
10 rows in set (0.00 sec)
```

Figure 4: Excerpt of Orders Table

In the above snapshot, the orders table is detailed. It can be seen that each order is identified using an orderID and the customer who placed the order is recognized using the customerID. For example, customer with ID=2 has placed an order which bears the ID=2.

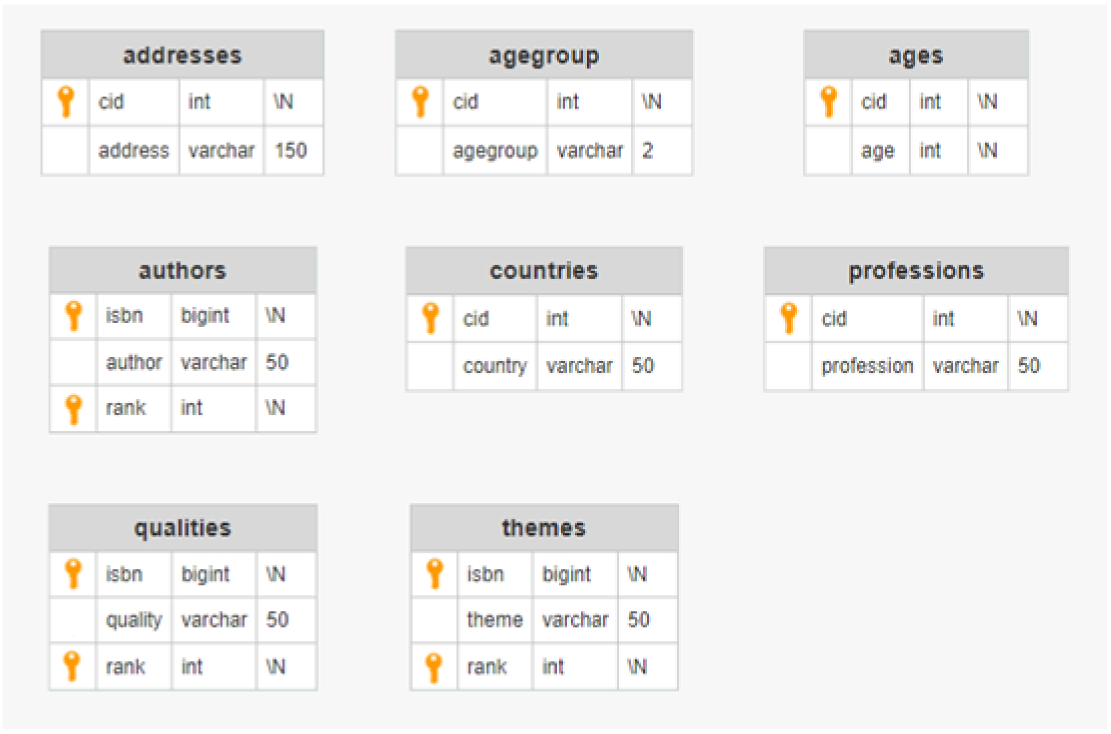


```
mysql> select a.cid,a.firstname,c.oid,b.title
    -> from customers a, titles b, orders c, ordersisbn d
    -> where a.cid=c.cid and c.oid=d.oid and d.isbn=b.isbn limit 5;
+-----+-----------+-----+-----------------------------+
| cid | firstname | oid | title                       |
+-----+-----------+-----+-----------------------------+
|   1 | TONY      |   1 | And Then There Were None    |
|   1 | TONY      |   1 | Stardust                    |
|   1 | TONY      |   1 | Mere Christianity           |
|   2 | THOR      |   2 | The Murder of Roger Ackroyd |
|   3 | BRUCE     |   3 | Murder on the Orient Express |
+-----+-----------+-----+-----------------------------+
5 rows in set (0.00 sec)
```

Figure 5: Order details retrieved by matching OID

In the snapshot provided above and using the orderID and customerID from the orders table, it can be seen that the customer with ID=2 is 'Thor' and has ordered the book with title 'The Murder of Roger Ackroyd'.

The gender of a customer is recorded in the genders table to provide recommendations after analyzing the purchases made by other customers belonging to the same gender.



addresses			
🔑	cid	int	\N
	address	varchar	150

agegroup			
🔑	cid	int	\N
	agegroup	varchar	2

ages			
🔑	cid	int	\N
	age	int	\N

authors			
🔑	isbn	bigint	\N
	author	varchar	50
🔑	rank	int	\N

countries			
🔑	cid	int	\N
	country	varchar	50

professions			
🔑	cid	int	\N
	profession	varchar	50

qualities			
🔑	isbn	bigint	\N
	quality	varchar	50
🔑	rank	int	\N

themes			
🔑	isbn	bigint	\N
	theme	varchar	50
🔑	rank	int	\N

Figure 6: Supporting tables of the Database

The ages table consists of field customerID and age. Both the fields are of type integer as they hold numeric value. The customerID is defined as the primary key and each customer's age is recorded in the corresponding field.

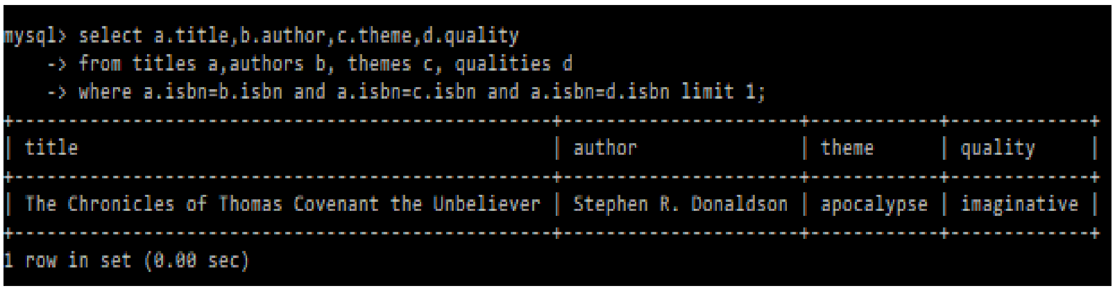
An addresses table is defined which hold the customerID and their corresponding addresses. The address field is of the type varchar as it holds bot numeric and textual data. Each of these addresses can be linked to the corresponding customer using the customerID which is the primary key.

The ageGroup table is created from the ages table to distinctly categorize all the customers present in the database. They are divided into groups signified by an alphabet. Customers below the age of 15 are placed in category A and customers above the age 15 but below 30 are placed in category B. On the same lines, customers above the age of 30 but below 50 are placed in category C and customers above the age of 50 are placed in category D. These categories are recorded in the ageGroup column of the table. Each customer record is identified by the customerID. This table is created to aid in recommending books based on the ordering behavior of customers belonging to the same age group.

The countries table hold the customerID and country fields. This table is used to record the customer's country of residence/origin to recommend books accordingly. The country field holds characters representing the name of the country. The customerID is defined as the primary key which helps in retrieval of specific information. The fields are not defined as a combined primary key as the design defines the customer to be a resident/origin of one single country.

The profession of the customers are recorded in the professions table which hold customerID and profession columns. The profession column is of the type varchar as it holds character data. The profession data is recorded to provide recommendation to customers based on the purchases of other customers who follow the same profession.

The authors table consists of ISBN, author, and rank columns. The columns are of the type bigint, varchar, and integer respectively. Each author is ranked as a few books may contain more than one authors. This leads to multiple records with same ISBN and hence a combined primary key consisting of ISBN and quality columns is used to uniquely identify each record present in the database. The authors table helps in filtering of books with the name of a specific author while providing recommendations to a customer.



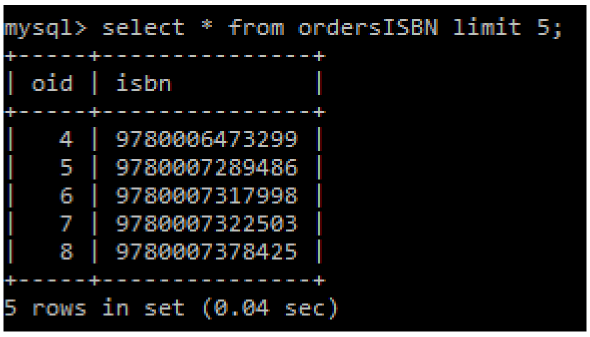
```
mysql> select a.title,b.author,c.theme,d.quality
    -> from titles a,authors b, themes c, qualities d
    -> where a.isbn=b.isbn and a.isbn=c.isbn and a.isbn=d.isbn limit 1;
+----------------------------------------------+---------------------+------------+-------------+
| title                                        | author              | theme      | quality     |
+----------------------------------------------+---------------------+------------+-------------+
| The Chronicles of Thomas Covenant the Unbeliever | Stephen R. Donaldson | apocalypse | imaginative |
+----------------------------------------------+---------------------+------------+-------------+
1 row in set (0.00 sec)
```

Figure 7: Book Details retrieved by matching ISBN

The qualities table consists of ISBN, quality, and rank columns. The columns are of the type bigint, varchar, and integer respectively. Each quality is ranked as a single book may contain various qualities. This leads to multiple records with same ISBN and hence a combined primary key consisting of ISBN and quality columns is used to uniquely identify each record present in the database. The qualities table helps in filtering of books with a specific quality while providing recommendations to a customer.

The themes table is similar to the qualities table and the only difference between the two being that the themes table ranks themes featured in the book instead of qualities. As a book can feature more than one themes, a combination of the ISBN and themes column is used as a primary key to identify each record uniquely. This table helps in providing recommendation to customers based on their theme preferences.

The database is in the first normal form as all tables in the database hold atomic values. The table is said to not be in 1NF if it violates the rule which states that: "each attribute of a table must have unique atomic values" (Singh, 2015). In the ordersISBN table below, it can be observed that each attribute of the table has single values.

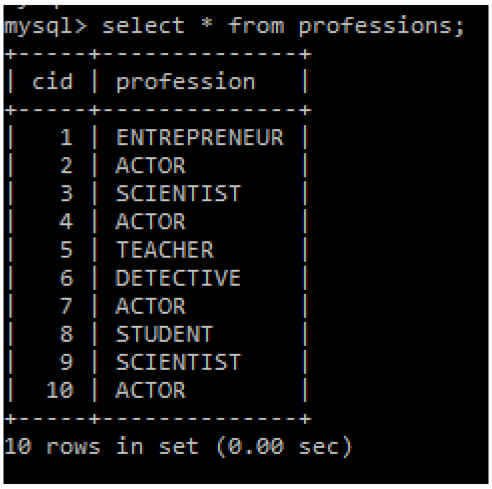


```
mysql> select * from ordersISBN limit 5;
+-----+---------------+
| oid | isbn          |
+-----+---------------+
|   4 | 9780006473299 |
|   5 | 9780007289486 |
|   6 | 9780007317998 |
|   7 | 9780007322503 |
|   8 | 9780007378425 |
+-----+---------------+
5 rows in set (0.04 sec)
```

Figure 8: Snapshot of ordersISBN table

In the image provided above it can be seen that the orderISBN table consists of orderID and ISBN columns. Each orderID is associated with an ISBN where the same orderID can have more than one values associated with it but an ISBN can be recorded only once in each orderID. The table stores these values such that each column has single values.

The tables in the database are mostly made up of two-three columns each where none of the non-prime attributes are dependent on the proper subset of any candidate key. For example, the customer details are split into ages, countries, genders, and professions table to ensure that the attributes (non-prime) contained within these columns can be stored in separate tables to ensure that the database is in the second normal form.

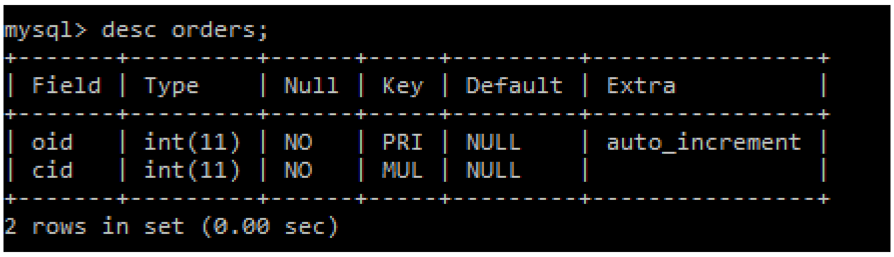


```
mysql> select * from professions;
+-----+--------------+
| cid | profession   |
+-----+--------------+
|   1 | ENTREPRENEUR |
|   2 | ACTOR        |
|   3 | SCIENTIST    |
|   4 | ACTOR        |
|   5 | TEACHER      |
|   6 | DETECTIVE    |
|   7 | ACTOR        |
|   8 | STUDENT      |
|   9 | SCIENTIST    |
|  10 | ACTOR        |
+-----+--------------+
10 rows in set (0.00 sec)
```

Figure 9: Snapshot of Professions table

In the above snapshot it can be observed that the profession of the customers are stored in a separate table as profession is a non-prime attribute but the table consists of customerID which is a primary key. This key can be used to retrieve the other non-prime attributes of the customer from other tables.

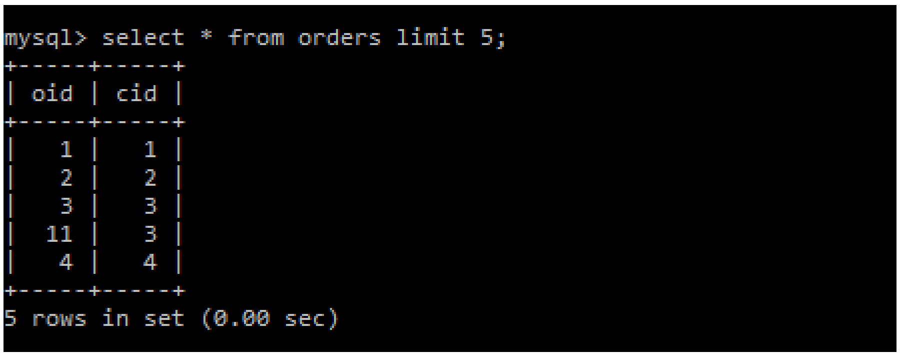
If the customer and order details can be stored within the same table there exists a transitive dependency where the order details are dependent on the orderID and the orderID is dependent on the customerID. In order to remove this transitive dependency the customer details and order details are stored separately. The orders table holds fields customerID and orderID to associate the order with the appropriate customer. The database is consistent with the third normal form rules.



```
mysql> desc orders;
+-------+---------+------+-----+---------+----------------+
| Field | Type    | Null | Key | Default | Extra          |
+-------+---------+------+-----+---------+----------------+
| oid   | int(11) | NO   | PRI | NULL    | auto_increment |
| cid   | int(11) | NO   | MUL | NULL    |                |
+-------+---------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

Figure 10: Structure of Orders Table

In the snapshot of orders table provided below, it can be seen that each customer has a unique orderID assigned to them for every individual order. However, a customer can place more than one orders and the same customer will then have more than one orderID. This can be observer for customer bearing the ID 3, where this particular customer has orderID 3 and 11 associated with themselves.



```
mysql> select * from orders limit 5;
+-----+-----+
| oid | cid |
+-----+-----+
|   1 |   1 |
|   2 |   2 |
|   3 |   3 |
|  11 |   3 |
|   4 |   4 |
+-----+-----+
5 rows in set (0.00 sec)
```

Figure 11: Snapshot of Structure and Contents of Orders table

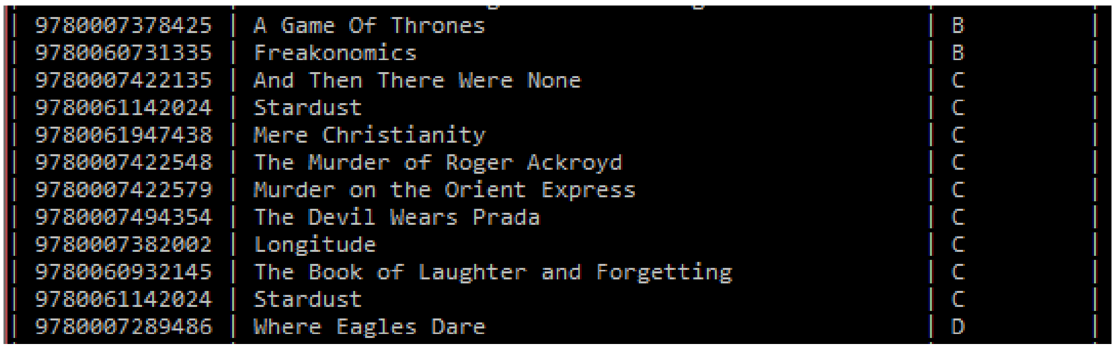
The database is in the Boyce Codd Normal Form as it complies with the third normal form and for every functional dependency, there exists a super-key. Each table is identified uniquely using the super-key. The customer details can be accessed using the customerID defined across the database. The order details can be accessed using the orderID and book details can be retrieved using the unique ISBN.

4. Database Views

The views provided onto the database provide insights to the type of data which is available to be used to provide recommendation. The views are used to combine data present across various tables in the database using complex select queries and present the result in a tabular form. They provide users access to only columns contained within the views and not to the underlying tables that is accessed by the view.

The database consists of views which contain information about the customer and books present in the database. Insights about the customer are provided using the ageGroupView, genderMale, genderFemale, countryView, and professionView. Information about books is provided in the themeView and qualityView. All these views can be constructed at real time using the stored procedures to update any changes that have occurred in the underlying tables.

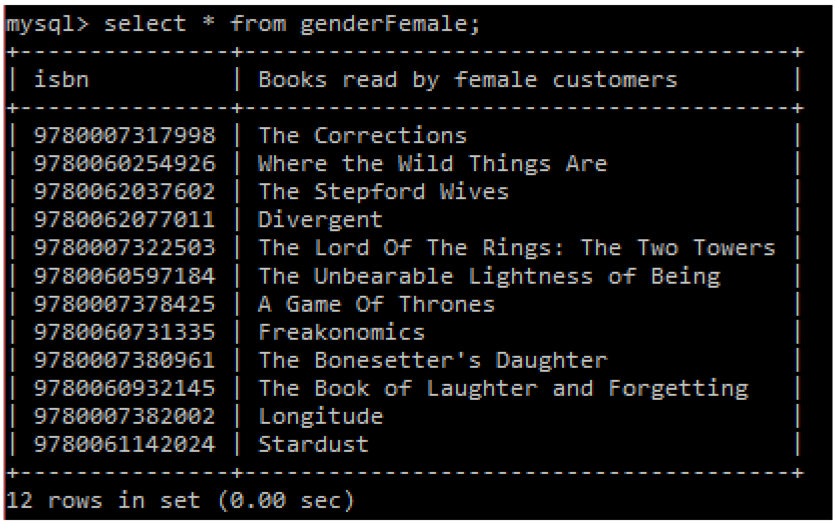
The ageGroupView consists of ISBN, title, and Age Group columns. The ISBN and title fields are retrieved from the titles table and the age-group of the customer is obtained from the ageGroup table. The customerID is matched from the ageGroup table and the orders table. The corresponding orderID from the orders table is matched to the orderID in the ordersISBN table to get the ISBN, and in turn, the book-title is retrieved from the titles table using the ISBN. This view can be used by both the customer and the user of the recommendation engine to understand the purchases of customers belonging to specific age-groups.



```
| 9780007378425 | A Game Of Thrones                             | B   |
| 9780060731335 | Freakonomics                                  | B   |
| 9780007422135 | And Then There Were None                      | C   |
| 9780061142024 | Stardust                                      | C   |
| 9780061947438 | Mere Christianity                             | C   |
| 9780007422548 | The Murder of Roger Ackroyd                   | C   |
| 9780007422579 | Murder on the Orient Express                  | C   |
| 9780007494354 | The Devil Wears Prada                         | C   |
| 9780007382002 | Longitude                                     | C   |
| 9780060932145 | The Book of Laughter and Forgetting           | C   |
| 9780061142024 | Stardust                                      | C   |
| 9780007289486 | Where Eagles Dare                             | D   |
```

Figure 12: Snapshot of ageGroupView

The above image is an excerpt from the ageGroupView. The ISBN, title of the book and the age-group of the customers who purchased the book is provided. For example, the book ‘A Game of Thrones’ is purchased by a customer belonging to age group ‘B’. This information can be used to provide recommendations to another customer belonging to the same age-group.

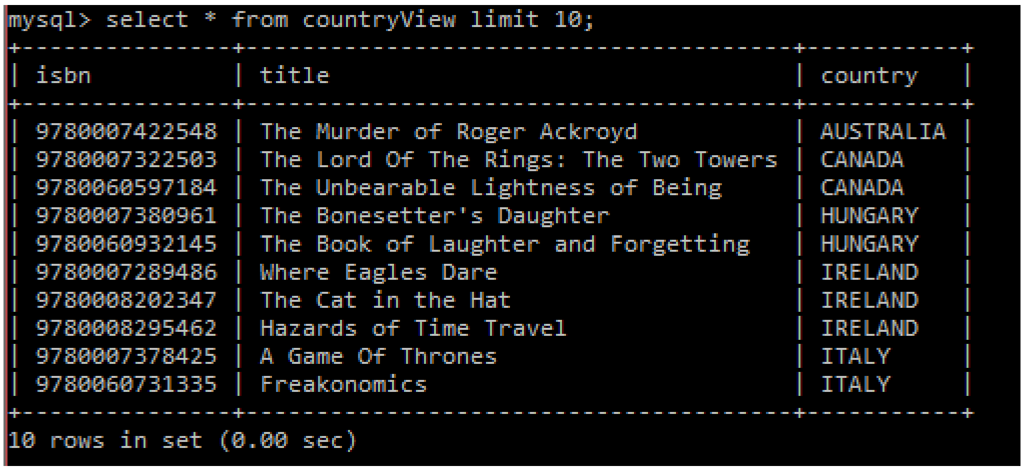


```
mysql> select * from genderFemale;
+---------------+----------------------------------------+
| isbn          | Books read by female customers         |
+---------------+----------------------------------------+
| 9780007317998 | The Corrections                        |
| 9780060254926 | Where the Wild Things Are              |
| 9780062037602 | The Stepford Wives                     |
| 9780062077011 | Divergent                              |
| 9780007322503 | The Lord Of The Rings: The Two Towers  |
| 9780060597184 | The Unbearable Lightness of Being      |
| 9780007378425 | A Game Of Thrones                      |
| 9780060731335 | Freakonomics                           |
| 9780007380961 | The Bonesetter's Daughter              |
| 9780060932145 | The Book of Laughter and Forgetting    |
| 9780007382002 | Longitude                              |
| 9780061142024 | Stardust                               |
+---------------+----------------------------------------+
12 rows in set (0.00 sec)
```

Figure 13: Snapshot of genderFemale view

The genderMale and genderFemale views are used to provide recommendations to customers by examining the gender that they belong to. These views provide the ISBN and title of the books that were ordered by customers of their respective genders. For example, in the screenshot given below, it can be seen that the book ‘Freakonomics’ was purchased by a female customer. This detail can be used while providing recommendations to another customer belonging to the same gender.

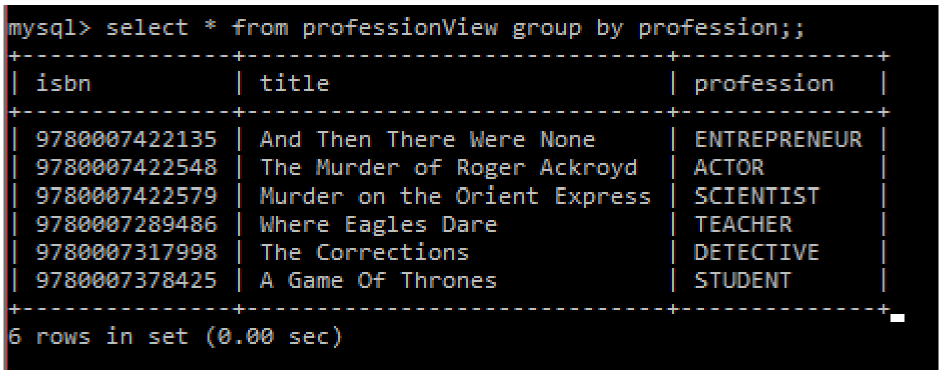
The countryView provides details of books that are ordered and the customer's country of origin/residence is associated with the name of the book. This can be used by the engine to provide location-based recommendations to a customer after analyzing the books that are ordered by other customers belonging to the same country. For example, the book 'Where Eagles Dare' was purchased by a customer from Ireland. This book can be suggested to another customer belonging to the same country.



```
mysql> select * from countryView limit 10;
+---------------+--------------------------------------+-----------+
| isbn          | title                                | country   |
+---------------+--------------------------------------+-----------+
| 9780007422548 | The Murder of Roger Ackroyd          | AUSTRALIA |
| 9780007322503 | The Lord Of The Rings: The Two Towers | CANADA    |
| 9780060597184 | The Unbearable Lightness of Being    | CANADA    |
| 9780007380961 | The Bonesetter's Daughter            | HUNGARY   |
| 9780060932145 | The Book of Laughter and Forgetting  | HUNGARY   |
| 9780007289486 | Where Eagles Dare                    | IRELAND   |
| 9780008202347 | The Cat in the Hat                   | IRELAND   |
| 9780008295462 | Hazards of Time Travel               | IRELAND   |
| 9780007378425 | A Game Of Thrones                    | ITALY     |
| 9780060731335 | Freakonomics                         | ITALY     |
+---------------+--------------------------------------+-----------+
10 rows in set (0.00 sec)
```

Figure 14: Snapshot of countryView

The professionView can be used to provide recommendations to customers belonging to a particular profession based on the purchasing behavior of other customers belonging to the same profession. This view provides the ISBN and title of the book along with the profession of the customer who made the purchase.

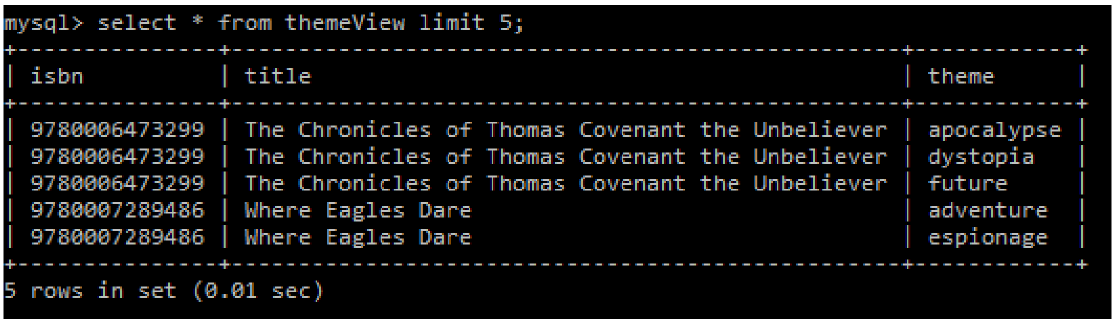


```
mysql> select * from professionView group by profession;;
+---------------+-----------------------------+--------------+
| isbn          | title                       | profession   |
+---------------+-----------------------------+--------------+
| 9780007422135 | And Then There Were None    | ENTREPRENEUR |
| 9780007422548 | The Murder of Roger Ackroyd | ACTOR        |
| 9780007422579 | Murder on the Orient Express | SCIENTIST    |
| 9780007289486 | Where Eagles Dare           | TEACHER      |
| 9780007317998 | The Corrections             | DETECTIVE    |
| 9780007378425 | A Game Of Thrones           | STUDENT      |
+---------------+-----------------------------+--------------+
6 rows in set (0.00 sec)
```

Figure 15: Excerpt of ProfessionView

The above image is an excerpt from the professionView and it can be observed that a customer who works as an entrepreneur ordered the book 'And Then There Were None'. This book can be suggested to another customer who has listed their profession as an entrepreneur.

For a new customer, the client-application can provide a list of themes and qualities to choose from and provide recommendation to the customer based on the chosen options. These options can be referenced with the themeView and qualityView which provide details of books possessing the enlisted feature. For example, if the customer chooses the theme of 'future', the book 'The Chronicles of Thomas Covenant the Unbeliever' can be suggested to the customer after referring the themeView.



```
mysql> select * from themeView limit 5;
+---------------+-----------------------------------------------+------------+
| isbn          | title                                         | theme      |
+---------------+-----------------------------------------------+------------+
| 9780006473299 | The Chronicles of Thomas Covenant the Unbeliever | apocalypse |
| 9780006473299 | The Chronicles of Thomas Covenant the Unbeliever | dystopia   |
| 9780006473299 | The Chronicles of Thomas Covenant the Unbeliever | future     |
| 9780007289486 | Where Eagles Dare                             | adventure  |
| 9780007289486 | Where Eagles Dare                             | espionage  |
+---------------+-----------------------------------------------+------------+
5 rows in set (0.01 sec)
```

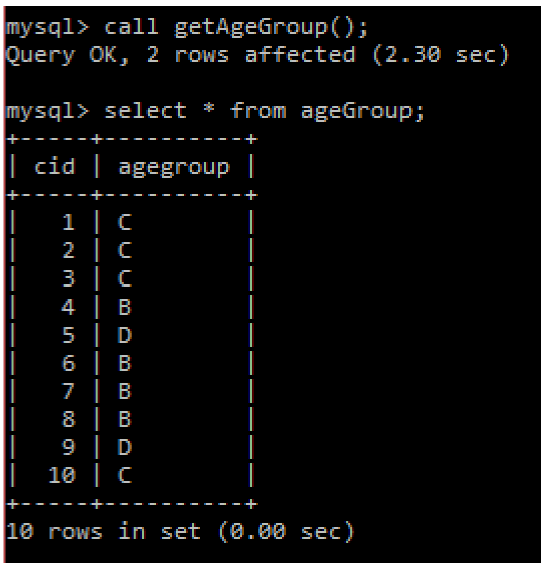
Figure 16Excerpt of the themeView

Views are mainly used to implement a level of security to the database. They provide an illusion of an SQL table to the user. The views display only specific data and any changes made to the view are not reflected in the underlying tables from which the view is constructed. Views can be used to display data aggregated from more than one tables. Any changes made to the underlying tables are also not reflected on the view. They are provide a consistent image of only the data which is available at the point of creation of the view. The views provide logical data independence which are otherwise enforced upon tables (Wadje, 2012). All these reasons project views as a good construct which can be used to represent certain SQL structures.

5. Procedural Elements

The database design employs stored-procedures and triggers. The stored procedures are used to provide built-in access control (6 Reasons Why You Should Use Stored Procedures, n.d.). The users interact with procedures and not directly with the table.

For example, the procedure getAgeGroup( ) is defined to create an ageGroup table which draws the customerID of the customer and categorizes them based on their age defined in the ages table. All this can be performed without allowing the user to directly access the customers and ages table. This procedure can be called each time a recommendation based on age-group needs to be made as it updates information of new users if entered.



```
mysql> call getAgeGroup();
Query OK, 2 rows affected (2.30 sec)

mysql> select * from ageGroup;
+-----+----------+
| cid | agegroup |
+-----+----------+
|   1 | C        |
|   2 | C        |
|   3 | C        |
|   4 | B        |
|   5 | D        |
|   6 | B        |
|   7 | B        |
|   8 | B        |
|   9 | D        |
|  10 | C        |
+-----+----------+
10 rows in set (0.00 sec)
```

Figure 17: AgeGroup Table created by a procedure

In the above image, the contents of the table ageGroup is displayed. The table is created using a stored procedure which categorizes the customers based on their ages.

Procedure enhance the reusability of code. In this database, they are used to create views at each procedure call which enables the views to be updated with current data. The following image provides the details of all the stored procedures present in the database.

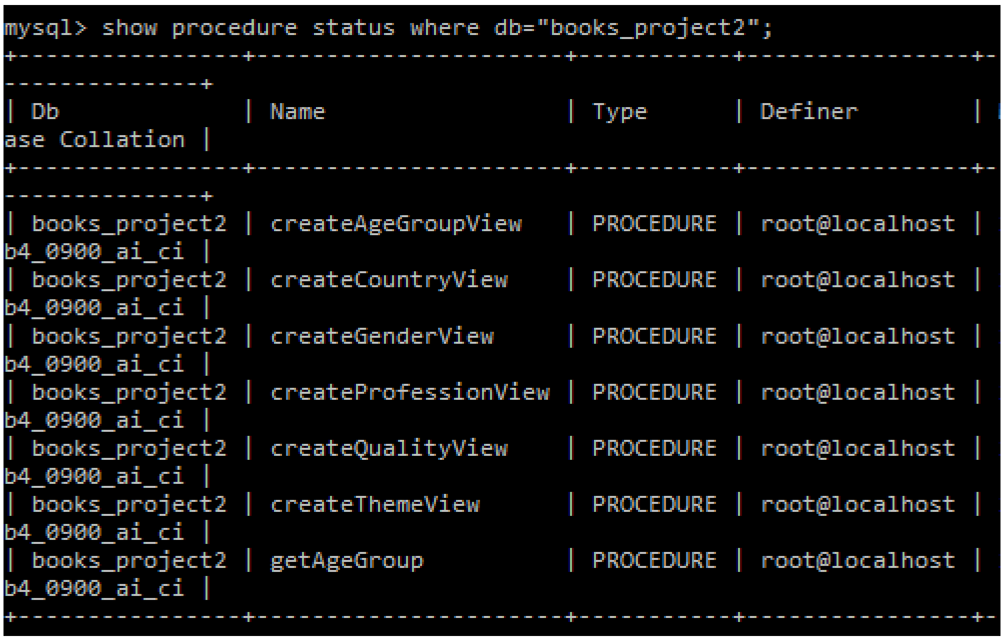

```
mysql> show procedure status where db="books_project2";
+----------------+---------------------+-----------+----------------+-
--------------+
| Db             | Name                | Type      | Definer        |
ase Collation |
+----------------+---------------------+-----------+----------------+-
--------------+
| books_project2 | createAgeGroupView   | PROCEDURE | root@localhost |
b4_0900_ai_ci |
| books_project2 | createCountryView    | PROCEDURE | root@localhost |
b4_0900_ai_ci |
| books_project2 | createGenderView     | PROCEDURE | root@localhost |
b4_0900_ai_ci |
| books_project2 | createProfessionView | PROCEDURE | root@localhost |
b4_0900_ai_ci |
| books_project2 | createQualityView    | PROCEDURE | root@localhost |
b4_0900_ai_ci |
| books_project2 | createThemeView      | PROCEDURE | root@localhost |
b4_0900_ai_ci |
| books_project2 | getAgeGroup          | PROCEDURE | root@localhost |
b4_0900_ai_ci |
+----------------+---------------------+-----------+----------------+-
```

Figure 18: Stored-Procedures defined in the Database

The createCountryView and createProfessionView are used to construct views containing details of books purchased by customers and the country and profession to which they are associated with. The createGenderView creates two views, each containing books read by one specific gender.

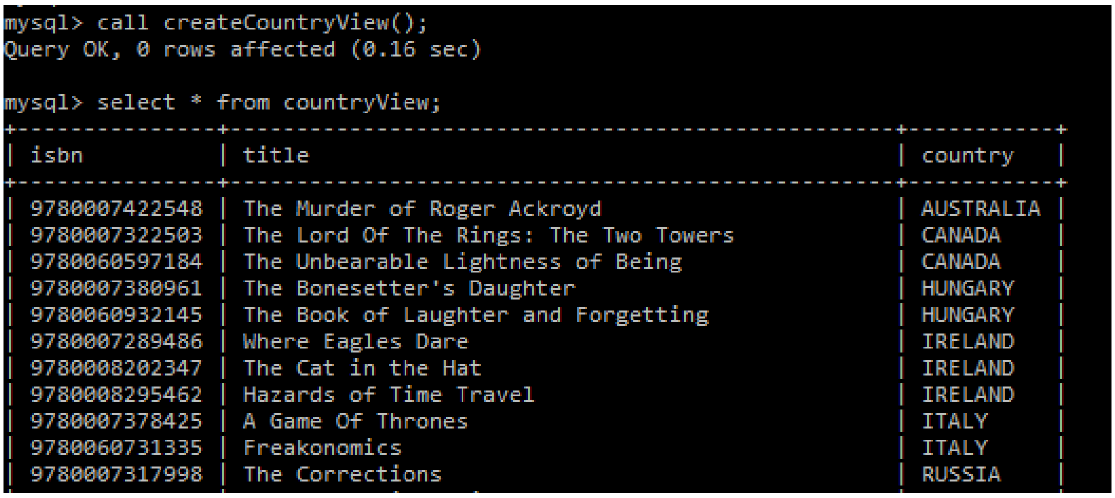
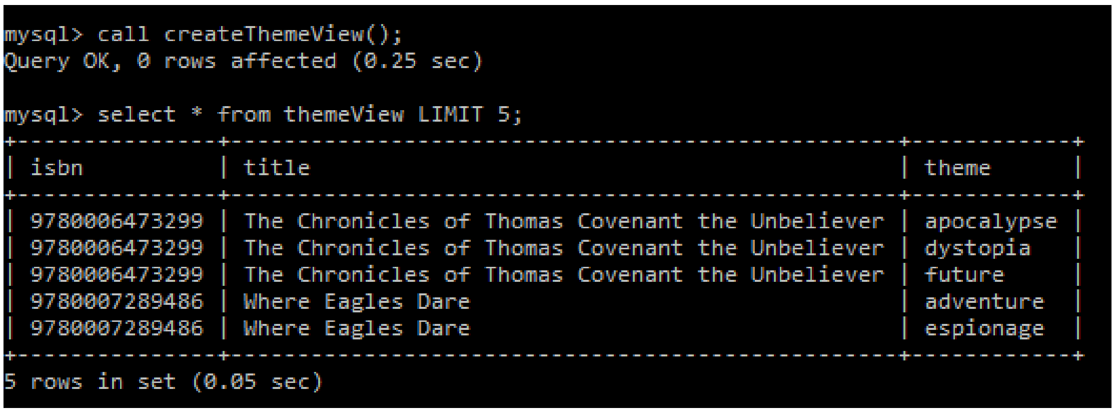

```
mysql> call createCountryView();
Query OK, 0 rows affected (0.16 sec)

mysql> select * from countryView;
+---------------+--------------------------------------------------+-----------+
| isbn          | title                                            | country   |
+---------------+--------------------------------------------------+-----------+
| 9780007422548 | The Murder of Roger Ackroyd                      | AUSTRALIA |
| 9780007322503 | The Lord Of The Rings: The Two Towers            | CANADA    |
| 9780060597184 | The Unbearable Lightness of Being                | CANADA    |
| 9780007380961 | The Bonesetter's Daughter                        | HUNGARY   |
| 9780060932145 | The Book of Laughter and Forgetting              | HUNGARY   |
| 9780007289486 | Where Eagles Dare                                | IRELAND   |
| 9780008202347 | The Cat in the Hat                               | IRELAND   |
| 9780008295462 | Hazards of Time Travel                           | IRELAND   |
| 9780007378425 | A Game Of Thrones                                | ITALY     |
| 9780060731335 | Freakonomics                                     | ITALY     |
| 9780007317998 | The Corrections                                  | RUSSIA    |
```

Figure 19: Excerpt of countryView

The above image shows an excerpt of the contents of the countryView. This view is created using the stored procedure createCountryView( ). This procedure is executed using the 'call' function along with the name of the procedure.

The createThemeView and createQualityView are used to provide details of books present in the database categorised according to the theme and quality that the book features.

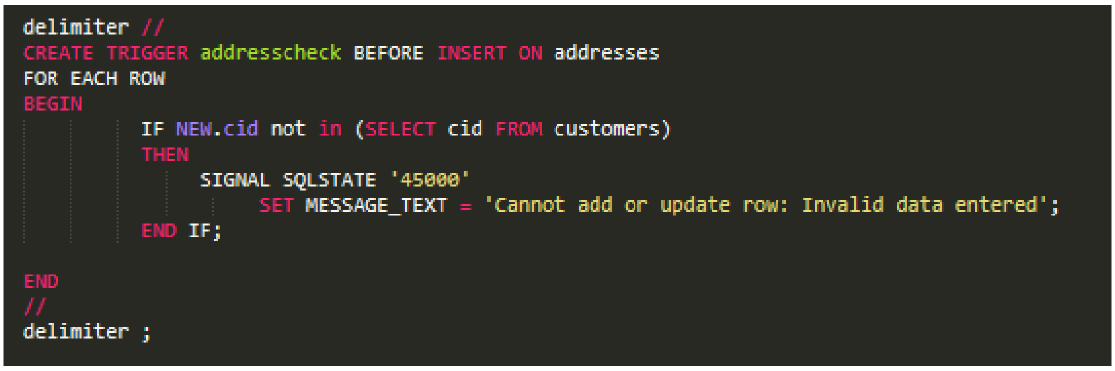


```
mysql> call createThemeView();
Query OK, 0 rows affected (0.25 sec)

mysql> select * from themeView LIMIT 5;
+---------------+-------------------------------------------------+------------+
| isbn          | title                                           | theme      |
+---------------+-------------------------------------------------+------------+
| 9780006473299 | The Chronicles of Thomas Covenant the Unbeliever | apocalypse |
| 9780006473299 | The Chronicles of Thomas Covenant the Unbeliever | dystopia   |
| 9780006473299 | The Chronicles of Thomas Covenant the Unbeliever | future     |
| 9780007289486 | Where Eagles Dare                               | adventure  |
| 9780007289486 | Where Eagles Dare                               | espionage  |
+---------------+-------------------------------------------------+------------+
5 rows in set (0.05 sec)
```

Figure 20: Excerpt of ThemeView

The createThemeView procedure create a view that contains the ISBN, title and the theme featured in the book. This view can be used to provide recommendations based on theme-preferences.

The database employs triggers to keep check on the consistency of data within the database. They are used to check the integrity of data being inserted into the database. Triggers are standard actions which are performed more than once by the program. They are useful to audit changes made to the data within tables. (Ahmed, 2015)



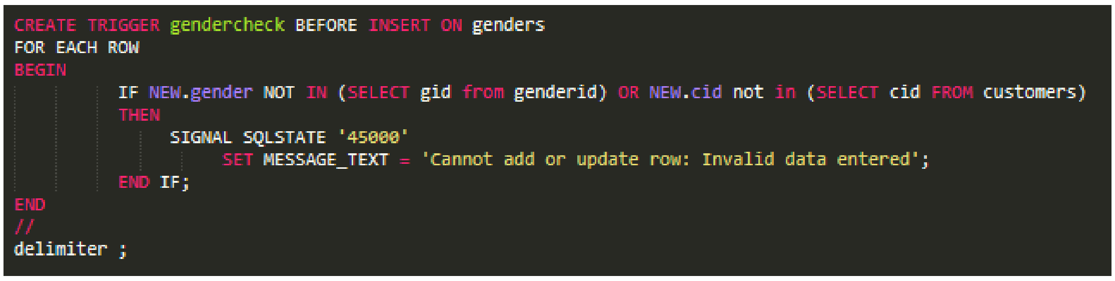
```
delimiter //
CREATE TRIGGER addresscheck BEFORE INSERT ON addresses
FOR EACH ROW
BEGIN
        IF NEW.cid not in (SELECT cid FROM customers)
        THEN
            SIGNAL SQLSTATE '45000'
                SET MESSAGE_TEXT = 'Cannot add or update row: Invalid data entered';
        END IF;

END
//
delimiter ;
```

Figure 21: addressCheck Trigger

Triggers such as addressCheck, countriesCheck and professionsCheck are employed to validate the data being inserted into the addresses, countries, and professions table respectively. The presence of the customerID being inserted is checked in the customers table. This ensures that only a valid customerID is being entered. If an attempt to enter an invalid customerID is made, an error signal is returned and a message communicating the action failure is displayed.

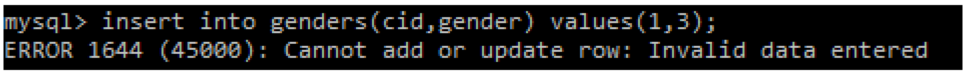
The ageCheck trigger is used to check the data being inserted into the ages table. A customer's age should be above five and the customerID must be a valid one that is present in the customers table. These constraints must be satisfied in order to enter data into the ages table, else, an error signal with a message is returned.

The genderIDCheck trigger is used to endure that no further records are added to the genderID table as it is designed to contain only two types of gender. The genderCheck trigger is used to check if the gender value being inserted into the genders table is a valid value that is present in the genderID table and also the customerID is validated with the customers table. The code of the genderCheck trigger is given below.



```
CREATE TRIGGER gendercheck BEFORE INSERT ON genders
FOR EACH ROW
BEGIN
        IF NEW.gender NOT IN (SELECT gid from genderid) OR NEW.cid not in (SELECT cid FROM customers)
        THEN
            SIGNAL SQLSTATE '45000'
                SET MESSAGE_TEXT = 'Cannot add or update row: Invalid data entered';
        END IF;
END
//
delimiter ;
```

Figure 22: genderCheck Trigger



```
mysql> insert into genders(cid,gender) values(1,3);
ERROR 1644 (45000): Cannot add or update row: Invalid data entered
```

Figure 23: genderCheck Trigger in action

In the above snapshot, an attempt to enter value '3' in the gender column is made and the action fails as the gender column is designed to hold either value 1 or 2 which is present in the genderID table.

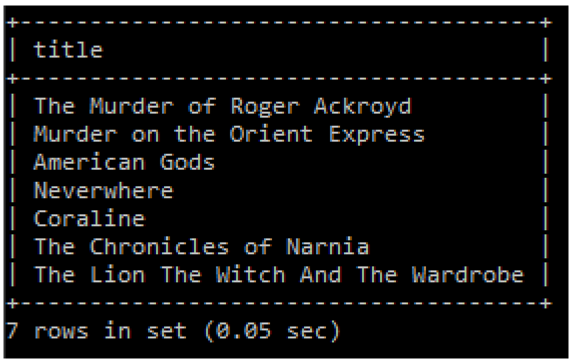
The themesIsbncheck, authorsIsbnCheck, and qualitiesIsbnCheck triggers are defined to ensure that only a valid ISBN is entered while updating the themes, authors, and qualities table respectively. The ISBN is validated by checking if it is present in the titles table before performing the update on the corressponding tables.

6. Example Queries

a) Query to find preferred books for a customer with ID=1 by analyzing the name of authors of the books purchased previously by the same customer:

```
select t.title from titles t, authors a
where t.isbn=a.isbn and a.author in
(select a.author from authors a, orders o1, ordersISBN o2
where o1.cid=1 and o1.oid=o2.oid and a.isbn=o2.isbn)
and t.title not in
(select t.title from titles t, orders o1, ordersISBN o2
where o1.oid=o2.oid and t.isbn=o2.isbn and o1.cid=1);
```

In this query, the customerID is taken into account and the name of authors of books purchased by the customer is retrieved initially. Titles of books (if any) authored by the authors obtained in the subquery is returned. The books which have been previously ordered by the customer is omitted from the results.



```
+------------------------------------+
| title                              |
+------------------------------------+
| The Murder of Roger Ackroyd        |
| Murder on the Orient Express       |
| American Gods                      |
| Neverwhere                         |
| Coraline                           |
| The Chronicles of Narnia           |
| The Lion The Witch And The Wardrobe |
+------------------------------------+
7 rows in set (0.05 sec)
```

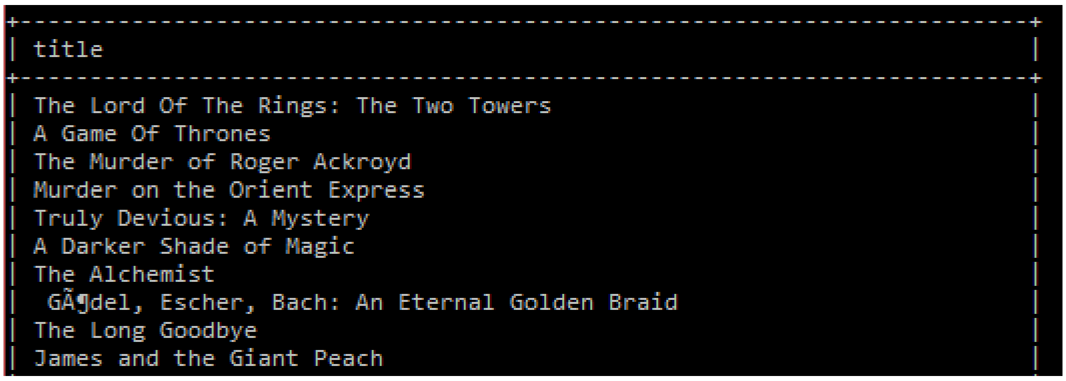
Figure 24: Query (a) Output

This query result can be used to provide customers item-based recommendation by examining the author names of the books that were ordered previously by the same customer. A customer might prefer to read more books written by the same author as an author.

b) Query to find book recommendation for customer with ID=1 based on the themes and qualities featured in previously ordered books by the same customer:

```
select distinct(d.title) from titles d, themes e
where d.isbn=e.isbn and e.theme in
(select a.theme from themes a, orders b, ordersISBN c
where b.cid=1 and b.oid=c.oid and c.isbn=a.isbn)
and d.title not in
(select a.title from titles a, orders b, ordersISBN c
where b.cid=1 and b.oid=c.oid and c.isbn=a.isbn)
and d.title in
(select distinct(d.title) from titles d, qualities e
where d.isbn=e.isbn and e.quality in
(select a.quality from qualities a, orders b, ordersISBN c
where b.cid=1 and b.oid=c.oid and c.isbn=a.isbn)
and d.title not in
(select a.title from titles a, orders b, ordersISBN c
where b.cid=1 and b.oid=c.oid and c.isbn=a.isbn));
```

The customerID is considered and the quality and theme featured in books that were previously ordered by the customer is returned by the subquery. The next subquery takes into account these features and returns titles based on them. The titles which have been ordered by the customer previously are not included in the results.



```
+------------------------------------------------------------+
| title                                                      |
+------------------------------------------------------------+
| The Lord Of The Rings: The Two Towers                      |
| A Game Of Thrones                                          |
| The Murder of Roger Ackroyd                                |
| Murder on the Orient Express                               |
| Truly Devious: A Mystery                                   |
| A Darker Shade of Magic                                    |
| The Alchemist                                              |
|  GÃ¶del, Escher, Bach: An Eternal Golden Braid             |
| The Long Goodbye                                           |
| James and the Giant Peach                                  |
```

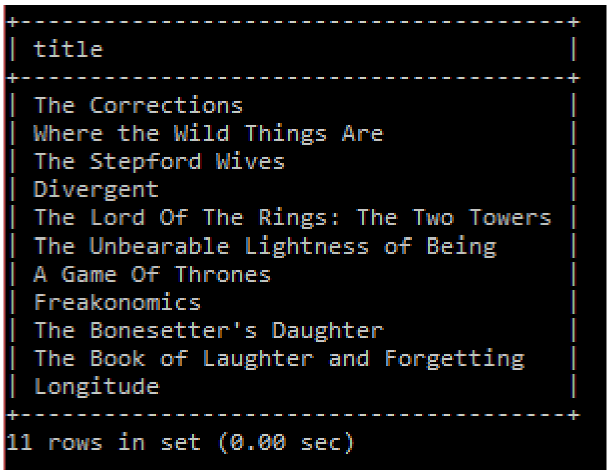
Figure 25: Excerpt of the Output of Query (b)

This query can be used to make recommendation to customers after analyzing the themes and qualities of previously ordered books. The suggestions retrieved contain books with similar themes and qualities which might appeal to the customer.

c) Query to provide recommendation using books that are ordered by customers belonging to a specific gender type:

```
select distinct(d.title) from titles d, ordersISBN e
where e.isbn=d.isbn and e.oid in
(select distinct(b.oid) from orders b, genders c
where b.cid=c.cid and c.gender in
(select a.gender from genders a
where cid=6))
and d.title not in
(select a.title from titles a, orders b, ordersISBN c
where b.cid=6 and b.oid=c.oid and c.isbn=a.isbn);
```

The customer chosen has an ID=6 and the records indicate that the customer is a female. This result is obtained in the subquery and books read by other female customers is returned. The books already ordered by the customer with ID=6 are not included in the results.



```
+--------------------------------------+
| title                                |
+--------------------------------------+
| The Corrections                      |
| Where the Wild Things Are            |
| The Stepford Wives                   |
| Divergent                            |
| The Lord Of The Rings: The Two Towers |
| The Unbearable Lightness of Being    |
| A Game Of Thrones                    |
| Freakonomics                         |
| The Bonesetter's Daughter            |
| The Book of Laughter and Forgetting  |
| Longitude                            |
+--------------------------------------+
11 rows in set (0.00 sec)
```

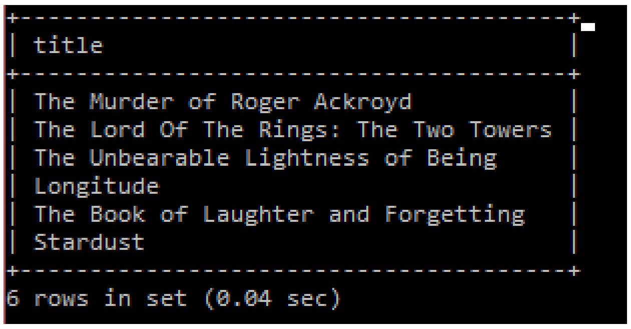
Figure 26: Output of Query(c)

The query results show recommendations which include books 'The Corrections' and 'Divergent'. This can be used to provide gender-specific recommendations by making user-based collaborative filtering which includes analyzing the ordering behavior of other customers belonging to the same gender type.

d) Query to find book recommendations for customers belonging to the same profession:

```
select d.title from titles d, orders e, ordersISBN f
where e.oid=f.oid and f.isbn=d.isbn and e.cid in
(select distinct(b.cid) from professions b, orders c
where b.cid=c.cid and b.profession in
(select a.profession from professions a
where cid=4))
and d.title not in
(select a.title from titles a, orders b, ordersISBN c
where b.cid=4 and b.oid=c.oid and c.isbn=a.isbn);
```

The subquery returns the profession of the customer bearing ID=4. The customer is an actor by profession and titles read by other customers who are actors by profession is returned.



```
+--------------------------------------+
| title                                |
+--------------------------------------+
| The Murder of Roger Ackroyd          |
| The Lord Of The Rings: The Two Towers |
| The Unbearable Lightness of Being    |
| Longitude                            |
| The Book of Laughter and Forgetting  |
| Stardust                             |
+--------------------------------------+
6 rows in set (0.04 sec)
```

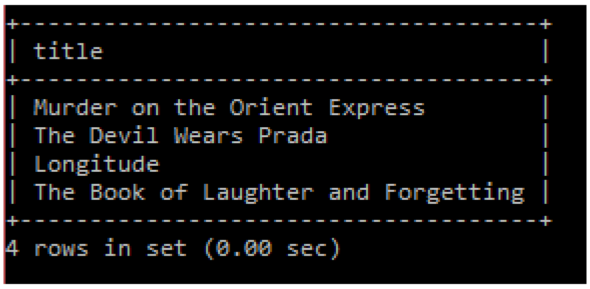
Figure 27: Output of Query (d)

The results of this query are for a customer who is an actor by profession. Similarly, recommendations can be made for customers by examining the books ordered by other customers belonging to the same profession (User-based collaborative filtering).

e) Query to find book suggestions based on location of a customer:

```
select d.title from titles d, orders e, ordersISBN f
where e.oid=f.oid and f.isbn=d.isbn and e.cid in
(select distinct(b.cid) from countries b, orders c
where b.cid=c.cid and b.country in
(select a.country from countries a
where cid=1))
and d.title not in
(select a.title from titles a, orders b, ordersISBN c
where b.cid=1 and b.oid=c.oid and c.isbn=a.isbn);
```

The location of a customer is identified by looking-up the relation containing the country details of a customer. Here, the customer bearing ID as 1 is recorded to be from USA. This value is returned from the subquery to return titles which have been ordered by other customers from USA. The previously ordered books of customerID=1 is not contained within the results.



```
+-------------------------------------+
| title                               |
+-------------------------------------+
| Murder on the Orient Express        |
| The Devil Wears Prada               |
| Longitude                           |
| The Book of Laughter and Forgetting |
+-------------------------------------+
4 rows in set (0.00 sec)
```

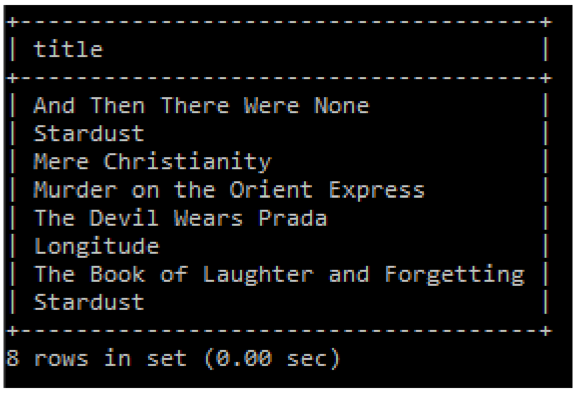
Figure 28: Output of Query (e)

This query can be used to provide recommendations based on the customer-location. The books ordered by other customers belonging to the same location are analyzed and suggested to a customer.

f) Query to find recommendations for customers belonging to a particular age group:

```
select d.title from titles d, orders e, ordersISBN f
where e.oid=f.oid and f.isbn=d.isbn and e.cid in
(select b.cid from agegroup b
where b.agegroup=
(select a.agegroup from agegroup a
where cid=2))
and d.title not in
(select a.title from titles a, orders b, ordersISBN c
where b.cid=2 and b.oid=c.oid and c.isbn=a.isbn);
```

The customer bearing ID=2 belongs to the ageGroup 'C'. This is retrieved by the subquery and the titles ordered by other customers of ageGroup 'C' is returned. These titles are filtered such that books previously ordered by customer with ID=2 is not returned.



```
+------------------------------------+
| title                              |
+------------------------------------+
| And Then There Were None           |
| Stardust                           |
| Mere Christianity                  |
| Murder on the Orient Express       |
| The Devil Wears Prada              |
| Longitude                          |
| The Book of Laughter and Forgetting |
| Stardust                           |
+------------------------------------+
8 rows in set (0.00 sec)
```

Figure 29: Output of Query (f)

It can be seen that a customer belonging to age group 'C' can be recommended with books such as 'And There Were None' and 'Stardust'. This query result can be used to recommend books that are ordered by other customers belonging to the same age group as a particular customer.

7. Conclusions

The database is designed specifically to support a book recommendation engine. However, the customer credentials can be integrated to other product recommendation systems as well. The decentralization of customer data means that additional information about a customer can be stored by creating new tables referenced by the customerID.

Customer data such as marital status, income, hobbies and interests can be stored in the database to provide more accurate recommendations to a customer after analyzing other customers who possess the same qualities. The database can be extended to accommodate a non-binary gender type as well. A more specific location of the customer can be utilized to provide more accurate recommendations. The address table is defined to accommodate this scope.

Feedback and ratings of a book can be obtained from customers who have made purchases. These ratings can be stored and used to provide recommendations to other customers who possess similar characteristics. The books can be tagged and stored based on languages to cater to a wider audience. The database facilitates this extension by allowing the creation of tables to accommodate more data about a specific book by referencing them with a common ISBN.

Acknowledgements

I would like to express my sincere gratitude to everyone who has provided me with means to complete this report successfully. I would like to thank my professor, Dr. Tony Veale, who was a source for knowledge and provided suggestions and constant encouragement which was essential to deliver this project. A special thanks goes to the teaching assistants and the lab demonstrators who were always willing to help solve any complication that was encountered while designing the database.

Furthermore, I would like to acknowledge that this work is entirely my own and every sentence written in this report is written by me and me alone. Any works referred to in preparation of this report have been cited inline.

References

6 Reasons Why You Should Use Stored Procedures. (n.d.). Retrieved from Kode Blog Tutorials: http://www.kode-blog.com/6-advantages-of-using-stored-procedures-explained

Ahmed, T. (2015, June 13). *Advantages and Disadvantages of Triggers in SQL*. Retrieved from Advance PHP: http://advancephp.com/advantages-and-disadvantages-of-triggers-in-sql/

Finnerty, S. (2017, September 11). *5 Steps to Setting up a Recommender System*. Retrieved from Klipfolio: https://www.klipfolio.com/blog/recommender-system

Ian. (n.d.). *MySQL Views*. Retrieved from Quackit: https://www.quackit.com/mysql/tutorial/mysql_views.cfm

Singh, C. (2015, May). *Normalization in DBMS: 1NF, 2NF, 3NF and BCNF in Database*. Retrieved from BeginnersBook: https://beginnersbook.com/2015/05/normalization-in-dbms/

Wadje, V. (2012, December 07). *Advantages and Disadvantages of views in SQL*. Retrieved from C# Corner: https://www.c-sharpcorner.com/blogs/advantages-and-disadvantages-of-views-in-sql-server1